
Clustering Words by Projection Entropy

Işık Barış Fidaner

Computer Engineering Department
Boğaziçi University, Istanbul
fidaner@alternatifbilisim.org

Ali Taylan Cemgil

Computer Engineering Department
Boğaziçi University, Istanbul
taylan.cemgil@boun.edu.tr

Abstract

We apply entropy agglomeration (EA), a recently introduced algorithm, to cluster the words of a literary text. EA is a greedy agglomerative procedure that minimizes projection entropy (PE), a function that can quantify the segmentedness of an element set. To apply it, the text is reduced to a feature allocation, a combinatorial object to represent the word occurrences in the text's paragraphs. The experiment results demonstrate that EA, despite its reduction and simplicity, is useful in capturing significant relationships among the words in the text. This procedure was implemented in Python and published as a free software: REBUS.

1 Introduction

Problems in natural language processing involve many difficulties due to the variety of subtleties and ambiguities in languages. On the other hand, these problems can be familiar to people in different fields, since the spoken and written languages constitute our common ground. It is especially favorable to approach these difficulties with generic statistical concepts, since these can also be utilized in other challenging fields such as bioinformatics. This paper demonstrates the use of entropy agglomeration (EA), a recently introduced algorithm [1], by clustering the words of a literary text.

By following the state-of-the-art NLP methods, we assume that *words* are the basic elements of a text [2, 3]. Moreover, to approach text analysis in a simpler form, we disregard all sequential ordering, considering each paragraph as a subset of words, and the whole text as a set of paragraphs. A statistical analysis of such data would conventionally be formulated in terms of joint probabilities of word sets to *co-occur* in the paragraphs. Such a probabilistic formulation can be extended to potentially infinite number of words by using Bayesian nonparametric models [4]. However, we take a different approach: we compute projection entropies (PE) of word sets, and use them in a clustering algorithm called entropy agglomeration (EA) to find the meaningful correlations among the words in the text. And we will briefly review the statistical concepts that were introduced in [1].

In the following sections, we define how the input data is represented by feature allocations, quantified by projection entropies and clustered by entropy agglomeration. We describe the experiment procedure on the way, and finally present the results, which demonstrate that our algorithm is able to capture a variety of meaningful relationships among the words of the input text. An additional discussion of projection entropy in comparison to co-occurrence is included in Appendix A.

2 The input text and its representation

We picked *Ulysses*¹ (1922) by James Joyce to be the input text. It consists of 7437 paragraphs and contains 29327 distinct words. But we only want to know which words occur in which paragraphs. This information is illustrated as a bipartite graph in Figure 1 where word elements below are linked to the paragraphs above them. Any text with n distinct words will be represented by a feature allocation defined as follows: A feature allocation of a set of elements $[n] = \{1, 2, \dots, n\}$, is a

¹Full text of this novel is available at <http://www.gutenberg.org/ebooks/4300>

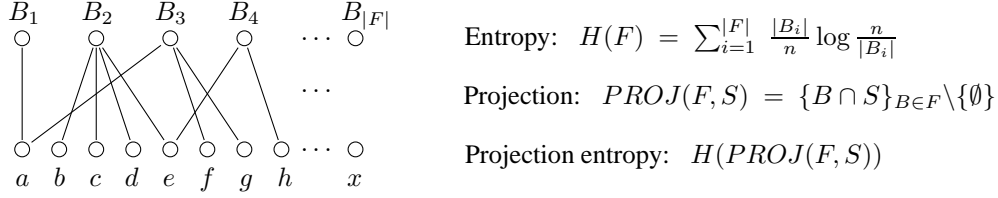


Figure 1: The text is represented by a feature allocation where each block represents a paragraph.

multiset of blocks $F = \{B_1, \dots, B_{|F|}\}$ such that $B_i \subset [n]$ and $B_i \neq \emptyset$ for all $i \in \{1, \dots, n\}$. The blocks $B_1, \dots, B_{|F|}$ in this definition will represent the paragraphs of the input text.

See the definitions given in Figure 1. Entropy quantifies the segmentedness of elements with respect to the blocks of F . It becomes maximum at the blocks that include about half of the elements. Projection of F onto S restricts the scope of F to this subset, functioning like a filter to focus on this particular subset. Projection entropy computes the segmentedness for a particular subset, by projecting F onto it (See [1]). If a subset has low PE, we say its elements have *entropic correlation*. The projection size $|PROJ(F, \{a\})|$ of an element a is the number of blocks that include it. In our case, a indicates a word and its projection size is the number of paragraphs it occurs in.

3 Clustering the word sets

Ulysses is reduced, but the feature allocation still contains too many words. Word sets of manageable sizes are needed for analysis. Nine word sets are assembled by restricting their projection sizes (number of paragraphs they occur in) in ranges 10, 11, 12-13, 15-17, 20-25, 30-39, 40-59, 60-149 and 150-7020. Then, the full feature allocation is projected onto each of these word sets, and EA is run on each of these projected feature allocations. Here is the pseudocode for the EA algorithm:

Entropy Agglomeration Algorithm:

1. Initialize $\Psi \leftarrow \{\{1\}, \{2\}, \dots, \{n\}\}$.
2. Find the subset pair $\{S_a, S_b\} \subset \Psi$ that minimizes the entropy $H(PROJ(F, S_a \cup S_b))$.
3. Update $\Psi \leftarrow (\Psi \setminus \{S_a, S_b\}) \cup \{S_a \cup S_b\}$.
4. If $|\Psi| > 1$ then go to 2.
5. Generate the dendrogram of chosen pairs by plotting minimum entropies for every bifurcation.

EA generates a dendrogram for each word set to show the entropic correlations among its elements. Dendrograms are diagrams commonly used to display results of hierarchical clustering algorithms [1, 5]. The dendrograms generated by this procedure for *Ulysses* can be examined in Appendix B.

Sample word pairs from EA dendrograms are shown in Table 1 to illustrate the variety of entropic correlations detected by the algorithm. These correlations indicate a diversity of semantic relationships: black-white, south-north are antonyms; then-now, former-latter, came-went are contraries;

Differences:	Inflections:	Expressions:		Figures:
black – white	his – he	ah – sure	window – seen	girl – sweet
south – north	her – she	ay – eh	hand – eyes	dame – joy
then – now	me – my	darling – perfume	face – head	females – period
former – latter	us – our	thank – please	cup – tea	wife – world
came – went	them – their	Things:	plate – fork	woman – behind
red – green	thy – thou	ocean – level	food – eating	gentleman – friend
female – male	hear – heard	waves – waters	job – business	gentlemen – friends
eve – adam	looking – looked	river – boat	sell – trade	priest – quietly
you – I	smile – smiled	moon – stars	slice – quantity	reverend – blessed
four – five	pouring – pour	birds – fly	family – memory	christ – jew
nine – eleven	ireland – irish	grass – fields	road – city	human – live
			system – distance	

Table 1: Sample word pairs to illustrate the entropic correlations captured by EA dendrograms

female-male, Eve-Adam, you-I indicate reciprocities; red-green are colors; four-five, nine-eleven are quantities; his-he, her-she, me-my, us-our, them-their, thy-thou are inflections of different pronouns; hear-heard, looking-looked, smile-smiled, pouring-poured are inflections of different verbs; ireland-irish is the inflection of a nation. Some of the other contextual correlations of expressions, things and figures are also enumerated. Entropic correlations cover an interesting range of meanings.

In conclusion, we developed in this work a text analysis tool that visualizes the entropic correlations among the words of a given text by applying entropy agglomeration to its paragraphs, and published this procedure as a free software: REBUS [6]. We demonstrated the utility of this procedure by clustering the words of a literary text using this tool.

Appendix A: On the meaning of projection entropy

Projection entropy (PE) is a useful guiding principle in exploring significant element-wise relationships in combinatorial data. But it has a meaning that is quite different from conventional statistical methods. Therefore in this section, we would like to discuss the meaning of projection entropy in comparison to a well-known quantity, *co-occurrence of elements*, which is used for similar purposes in probabilistic modeling. The actualizations of these quantities' values for 2 and 3 elements are illustrated in Figure 2. Firstly, as pointed out in [1], contrary to the positive sense of co-occurrence as scoring the blocks where all elements co-occur; PE has a negative sense of penalizing the blocks that divide and separate them. Secondly, co-occurrence is non-zero only at the blocks that include all of the elements, leaving out the blocks that exclude any of them. But PE leaves out the blocks that include all elements as well as the blocks that exclude all elements; it is non-zero only at partial inclusions: at the blocks that include some elements while excluding other elements. This makes PE a flexible quantity that can adjust to the blocks where any part of the elements overlap, whereas co-occurrence is a rigid quantity that can only adjust to the blocks where all of the elements overlap.

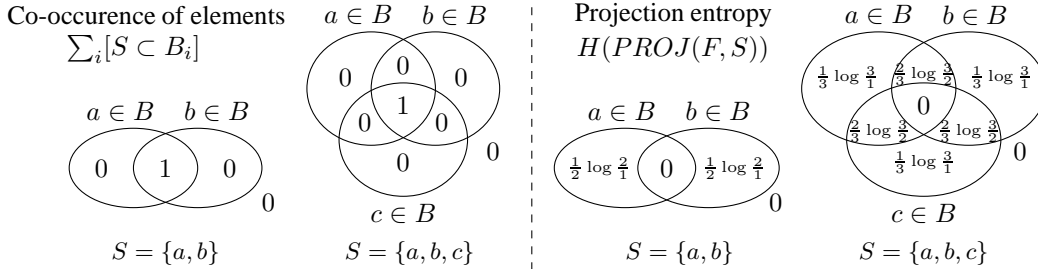
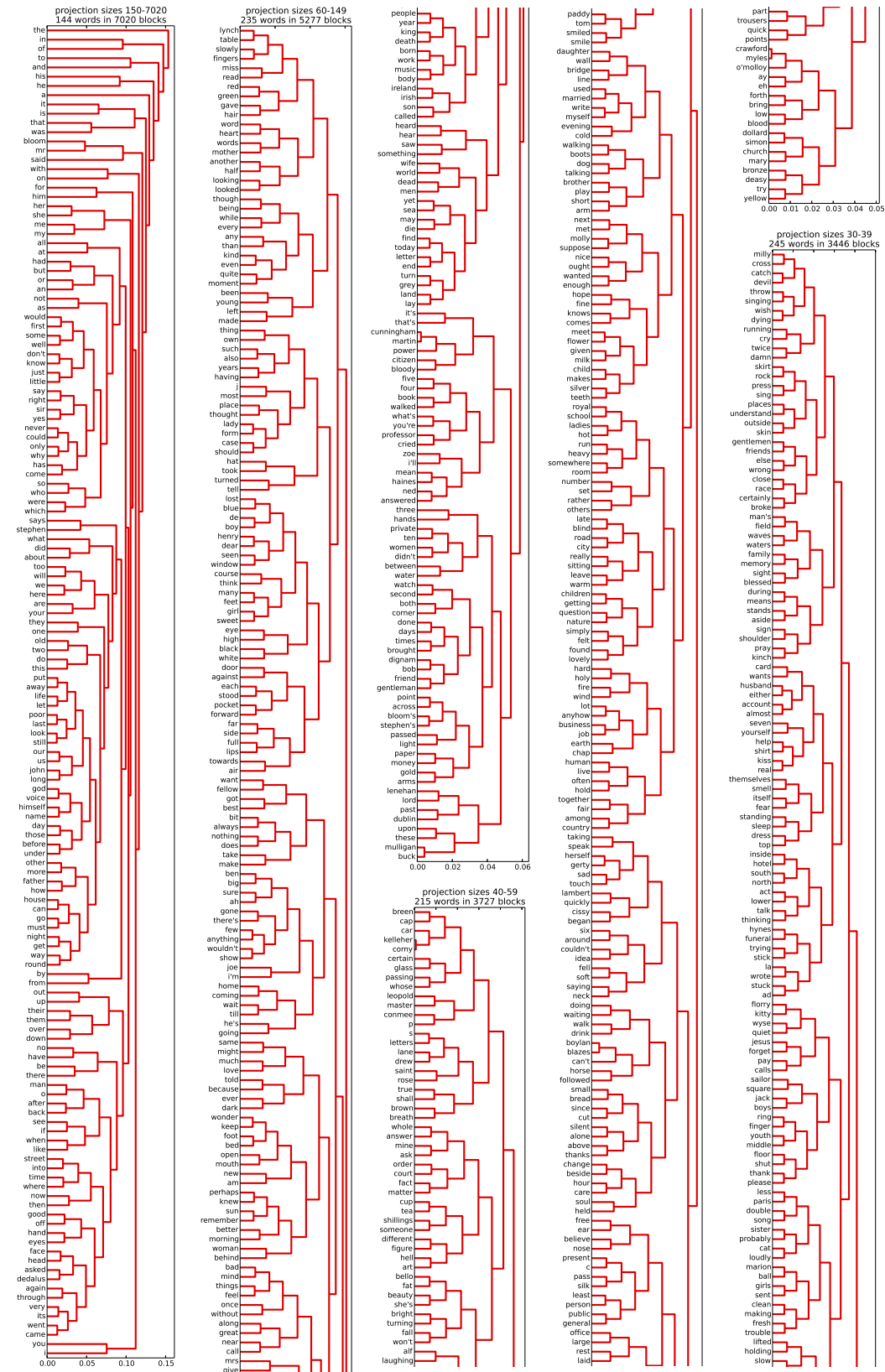


Figure 2: Comparing co-occurrence and projection entropy in their values for 2 and 3 elements

Assume that we have a cluster S whose elements have constant projection sizes. If these elements do not overlap at any of the blocks that include any of them, the cluster's PE will take the maximum value: the sum of projection sizes multiplied by $\frac{\log |S|}{|S|}$. If these elements overlap at all of the blocks that include them, PE will be zero by definition. Moreover, any additional overlapping among the elements will decrease the cluster's PE, if the projection sizes are kept constant. Therefore, lower PE indicates higher overlapping in the cluster, relative to its elements' projection sizes. To express this element-wise overlapping indicated by a lower PE, we say that these elements have an *entropic correlation* at the blocks that include them.

To understand how PE functions in entropy agglomeration, let us examine its effect for a word pair. Assume that we have a pair from the set of words that occur exactly in 10 paragraphs. We know that (1) projection sizes for both words are 10, (2) co-occurrence would count the blocks that include both of them, (3) PE would count the blocks that include one of them. For this particular case, these two quantities are directly proportional: an increase in co-occurrence by 1 would decrease PE by $\log 2$. This makes them practically equivalent. However, if there are several projection sizes like 20-25, words with larger projections can have partial occurrences more frequently; co-occurrence would ignore these occurrences, but PE may count them to penalize the elements for occurring 'unnecessarily'.

Appendix B: Entropy agglomeration dendrograms for *Ulysses*²



² Full results of this experiment and the Python code can be found on the REBUS website [6].

References

- [1] Fidaner, I. B. & Cemgil, A. T. (2013) Summary Statistics for Partitionings and Feature Allocations. In *Advances in Neural Information Processing Systems*, 26.
- [2] Wood, F., Archambeau, C., Gasthaus, J., James, L. F. & Teh, Y.W. (2011) The Sequence Memoizer. *Communications of the ACM*, 54(2):91-98.
- [3] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. (2013) Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, 26.
- [4] Teh, Y. W. (2006) A hierarchical Bayesian language model based on Pitman-Yor processes. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (ACL-44). Association for Computational Linguistics, Stroudsburg, PA, USA, 985-992.
- [5] Eisen, M. B., Spellman, P. T., Brown, P. O., & Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863-14868.
- [6] Fidaner, I. B. & Cemgil, A. T. (2014) REBUS: entropy agglomeration of text. Published under GNU General Public License. Online: <http://fidaner.wordpress.com/science/rebus/>